

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student:

Tomáš Kaňok

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: Yellen solutions s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti
 - c) Zvolený postup řešení zadaných úkolů
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vedl odbornou praxi studenta.


Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Zdeněk Sawa, Ph.D.**


Konzultant bakalářské práce: Rostislav Přidal

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013


doc. Dr. Ing. Eduard Sojka
vedoucí katedry




prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 28. dubna 2013

Tomáš Láňal

Rád bych zde poděkoval panu Rostislavu Přidalovi za umožnění odborné praxe ve firmě YELLEN solutions s.r.o. Poděkování také patří Mgr. Martinu Hapalovi, který ve firmě se mnou řešil většinu projektů. Nezapomínám ani na další spolupracovníky, se kterými jsem měl tu čest konzultovat a rozebírat problémy. Děkuji také vedoucímu bakalářské práce Ing. Zdeňkovi Sawovi, Ph.D.

Abstrakt

Tato bakalářská práce popisuje praxi, kterou jsem absolvoval ve firmě YELLEN solutions s.r.o. na pozici programátora. Během ní jsem pracoval na několika webových projektech, naprogramoval dvě menší desktopové aplikace a podílel jsem se na vývoji rozsáhlého informačního systému.

V úvodu své práce nejdříve představuji firmu, v níž praxe proběhla. Následuje část obsahující seznam projektů, na kterých jsem pracoval, jejich popis a řešení. Na konci naleznete zhodnocení mé praxe a zkušenosti, které jsem získal.

Klíčová slova: Webové aplikace, Java, C#, SQL, OOP, UML, párové programování, widget, framework

Abstract

This bachelor thesis describes a practical training I have undergone in the company Yellen solutions s.r.o. as a computer programmer. I worked on several web projects, programmed two minor desktop applications and participated in a development of an extensive information system.

In the introduction of the thesis I introduce the company in which I have undergone the practical training. The following part includes the list of projects I have worked on, their descriptions and solutions. In the final part the evaluation of the practical training and the experience I have gained are presented.

Keywords: Web applications, Java, C#, SQL, OOP, UML, pair programming, widget, framework

Seznam použitých zkratk a symbolů

CSS	–	Cascading Style Sheets
GUI	–	Graphical User Interface
LCD	–	Liquid Crystal Display
SQL	–	Structured Query Language
SSH	–	Secure Shell
SVN	–	Apache Subversion
XML	–	Extensible Markup Language

Obsah

1	Úvod	4
2	Popis firmy	5
2.1	Hlavní projekty	5
2.1.1	Laňka	5
2.1.2	Wapiti	6
2.2	Moje pracovní zařazení	6
3	Řešené úkoly a projekty během praxe	7
3.1	Quick programming	7
3.1.1	Vytvoření webového rozhraní a administrace	7
3.2	Laňka	8
3.2.1	Přestylování aplikace na Samsung Galaxy S	8
3.2.2	Widget Slevy	8
3.3	Dotazník	10
3.4	Rezervační systém pro ubytovnu Pocheň	11
3.5	Nasazení webové prezentace	13
4	Závěr	14
4.1	Uplatněné znalosti ze studia na vysoké škole	14
4.2	Získané znalosti při praxi	14
4.3	Scházející znalosti v průběhu praxe	14
4.4	Zhodnocení praxe	14
5	Reference	15

Seznam obrázků

1	Časová náročnost objednávky	5
2	Propojení tabulek používané widgetem Slevy	9

Seznam výpisů zdrojového kódu

1	Ukázka metody pro načítání dat z databáze	12
---	---	----

1 Úvod

Během studia na fakultě elektrotechniky a informatiky na Vysoké škole báňské - Technické univerzitě Ostrava jsem řešil několik úkolů a projektů. Nikdy to nebylo něco komplexnějšího, snad díky časové náročnosti. Měl jsem i pochybnost, zda se opravdu dané postupy, které jsem se naučil většinou v teoretické rovině, dají v reálné praxi použít. Z tohoto důvodu jsem chtěl absolvovat takovou odbornou praxi, která by mi dala nejen reálnější pohled, ale i cenné praktické zkušenosti v této oblasti. Zároveň jsem hledal příležitost pracovat v kolektivu více lidí a konzultovat s nimi dané postupy a problémy.

Dostala se mi možnost vykonávat bakalářskou praxi ve firmě YELLEN solutions s.r.o. a to rovnou na pozicích, které mě už od střední školy zajímají. Jedná se o vývoj a realizaci velkých informačních systémů, webových aplikací a databází.

V bakalářské práci nejdříve představím firmu. Následuje popis vykonávaných prací a řešení jednotlivých úkolů, na kterých jsem se podílel. Na závěr subjektivně zhodnotím, do jaké míry jsem využil zkušenosti a dovednosti získané během studia na Vysoké škole báňské – Technické univerzity Ostrava.

2 Popis firmy

Společnost YELLEN solutions s.r.o. vznikla se záměrem pořádání různých, především adrenalinových, teambuildingových akcí. Postupně se její zaměření měnilo a vyvíjelo. Nyní se zaměstnanci firmy plně věnují informačním technologiím. Stěžejní náplní práce firmy je vývoj vlastního informačního systému, vývoj systému na řízení obrazovek a nedílnou součástí jejich náplně je tvorba webových aplikací. Rovněž vyvíjí software pro různá zařízení. Navíc poskytuje pro své klienty vlastní webhosting.

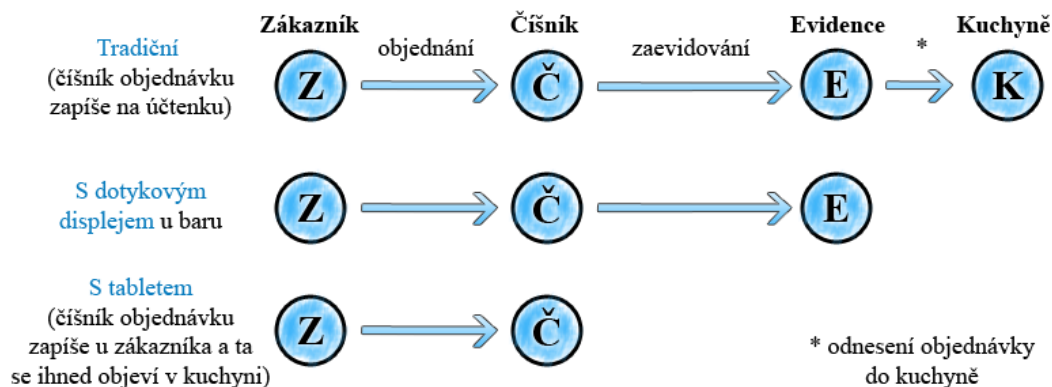
2.1 Hlavní projekty

Kromě různých webových aplikací firma vyvíjí vlastní informační systém nazvaný Laňka a systém na řízení obrazovek Wapiti.

2.1.1 Laňka

Laňka je informační systém, který pomáhá s řízením restaurací, barů, čajoven a dalších podobných zařízení. Systém lze zobrazit na více zařízeních. Dá se použít stolní počítač, tablet, chytrý mobil nebo pokladna. Vše je mezi sebou propojené pomocí kabelu nebo bezdrátově přes Wi-Fi.

Například servírka nebo číšník mohou posílat objednávky přímo od stolu zákazníka do kuchyně. Už se nemůže stát, že se objednávka vystavená v papírové podobě někde v kuchyni ztratí – zapadne a host marně čeká na její vyřízení. Objednávky se řadí časově za sebou tak, jak si hosté objednávají, a v tomto pořadí jsou také jednotlivé pokrmy připravovány. Obsluha tímto získá více času a může ho věnovat hostům. Systém pomáhá k rychlejší komunikaci: host – obsluha – kuchyň a tím ke zkvalitnění práce obsluhy.



Obrázek 1: Časová náročnost objednávky

Nechybí ani další doplňkové funkcionality jako inventarizace, vyúčtování a další. Ty jsou řešeny jako doplňky a přidávají se do systému podle přání a požadavků klienta.

Co se týče programovacího jazyku, tak se systém vyvíjí v objektovém orientovaném jazyku Java. Pro vývoj se používá vývojové prostředí Eclipse. Do něj se musí doinstalovat Google plugin pro Eclipse [5], což je skupina nástrojů pro vývoj webových aplikací. Tento nástroj je Googlem vyvíjen jako open-source. Samozřejmě informační systém používá také databázi, která je v tomto případě MySQL. Firma používá k tomuto účelu XAMPP. Pro správu a verzování zdrojových kódů se používá TortoiseSVN klient. Při příchodu do práce bylo tedy nutné vždy provést update. Pro připojení k testovacímu serveru je zapotřebí programu PuTTY a jeho SSH agenta Pageant.

2.1.2 Wapiti

Wapiti je systém na řízení LCD obrazovek. Produkt má webové rozhraní, kdy si klient může navolit požadované funkcionality obrazovky. To znamená, kdy a jakou formou se mají na obrazovce zobrazit reklamní spoty nebo kdy se má obrazovka automaticky zapnout či vypnout. K tomuto produktu firma zajišťuje také hardware, jeho instalaci a podporu v podobě servisu a poradenství.

2.2 Moje pracovní zařazení

Po přijetí na odbornou praxi jsem byl součástí tříčlenného vývojového týmu. Mé zařazení bylo na pozici programátora. Mým úkolem bylo programování různých aplikací nebo jejich částí, které většinou nemusely být ihned vyvinuty pro okamžité používání. Měl jsem tedy dostatek času na splnění zadaných úkolů.

3 Řešené úkoly a projekty během praxe

Úkoly mi byly zadávány osobně při konzultacích a přes internet. Firma pro řízení projektů používá webovou službu nazvanou Redmine [2]. Zde jsem měl k projektu zadané všechny potřebné informace rozdělené do menších – dílčích úkolů, které jsem postupně řešil a konzultoval s ostatními spolupracovníky. Aplikace taky nabízí kontrolu a komentování jednotlivých úkolů, časový průběh a další funkcionality.

3.1 Quick programming

3.1.1 Vytvoření webového rozhraní a administrace

Popis

Mým prvním projektem bylo naprogramování webové služby Quick programming. Dostal jsem do Redmine systému popis, co všechno by měla aplikace obsahovat. Jedná se o stránky, kde návštěvník (zákazník) přijde a objedná si přes formulář naprogramování nějakého skriptu. Důraz zde byl kladen na rychlou odezvu na objednávku, okamžité programování a odevzdání v řádu pár hodin.

Řešení

Nejdříve jsem musel s grafikem vyjednat novou grafickou stránku webové prezentace. Dosavadní už nevyhovovala rychle se měnícímu vývoji trhu. Grafikovi jsem zároveň poslal i své náčrtky, jak by mohly být prvky na stránce poskládány a co všechno by měla stránka obsahovat. Po několika úpravách jsme došli k závěru, který se líbil jak mně, tak i firmě YELLEN solutions s.r.o. Zároveň jsem vypracoval UML sekvenční diagram pro lepší představu, jak bude služba Quick programmingu pracovat.

Ve výběru programovacího jazyku jsem nebyl nijak omezen. Nakonec jsem zvolil čisté PHP bez frameworků. Pro lepší interakci s uživatelem jsem použil javascript a framework jQuery. Jelikož v grafice používané písmo nebylo standardní, byla by velká šance, že dané písmo uživatel na svém zařízení nemá. Rozhodl jsem se tedy použít javascriptovou knihovnu Cufón [3]. Nejprve bylo potřeba daný font nechat zpracovat generátorem, který je na stránkách knihovny. Pak již bylo možné písmo použít.

Aplikace také měla obsahovat jednoduchou administraci pro spravování zakázek. Jako databázový systém jsem mohl zvolit MySQL nebo PostgreSQL. Použil jsem první zmíněný, jelikož jsem s ním měl už zkušenosti z projektů na vysoké škole. Při vývoji jsem také využil knihovnu LESS, která slouží k usnadnění psaní CSS souborů. Umožňuje používání proměnných, jednoduchých matematických operací a funkcí v kaskádových stylech. Po všech úpravách lze vygenerovat z LESS knihovny čistý CSS soubor. Není tedy potřeba dávat na webhosting další knihovnu, která by jej jen zpomalovala.

3.2 Laňka

3.2.1 Přestylování aplikace na Samsung Galaxy S

Popis

Mým dalším úkolem byla změna CSS stylů pro rezervační systém Laňka. Aplikace byla vyvíjena pro větší rozlišení na počítače a tablety. Já jsem měl za úkol vytvořit korektní zobrazování i pro rozlišení 480 x 800 pixelů, které má chytrý mobilní telefon Samsung Galaxy S. Měl jsem na výběr ze dvou způsobů úprav. A to upravovat přímo soubor se styly nebo změnit generování CSS stylů v zdrojovém kódu.

Řešení

Nejprve jsem se snažil provést změnu stylování přímo v informačním systému. Zahrnovalo to pochopení velkého množství Java zdrojových kódů. Tento způsob se ukázal být pracný a velice neefektivní. Proto jsem rozhodl upravovat přímo soubor se styly, který obsahoval přes 1200 řádků. K lepší orientaci jsem použil Google Chrome a jeho Developer Tools. Abych mohl simulovat na počítači přesně zadané rozlišení 480x800, byl mi doporučen plugin Web Developer [4]. K účelům testování mi byl propůjčen i samotný Samsung Galaxy S.

Úkol byl náročný v tom, že jsem musel projít úplně všechny možnosti a funkcionality.

3.2.2 Widget Slevy

Díky tomuto úkolu se dostal až k samotnému programování informačního systému Laňka. Abych měl seznámení se složitým vývojem snazší, domluvil jsem se na párovém programování. Tato technika spočívá v tom, že dva vývojáři pracují na úkolu společně. Jeden přímo vytváří kód na počítači, druhý přemýšlí o prováděných změnách. Toto má hned několik výhod. Vývojáři řeší problémy společně, takže se na řešení přijde rychleji. Navíc se při tvoření kódu vzájemně kontrolují a tím pádem nedochází tak často k logickým chybám.

Vzájemná komunikace napomáhá k většímu soustředění. Slouží to také ke zvýšení kvalifikace méně zdatného programátora. V tomto případě jsem to byl já. Spolupracovník byl opravdu vstřícný a na každou mou otázku mi odpověděl a dokázal mi problém srozumitelně vysvětlit.

Popis

Widget má umět více druhů slev:

- **Pevná sleva** - velikost slevy je přesně určena
- **Procentuální sleva** - cena produktu je zmenšena o procentuální podíl

Widget má umět více druhů slev. Sleva může být sleva daná pevnou částkou nebo procentem. V informačním systému musí být možnost přiřadit každého zákazníka do

Další problém nám nastal při tvoření tabulky slev. Jelikož typů slev může být více, bylo nutné ji promyslet. Pro každý nový druh slevy se mohla vytvořit nová tabulka. Toto však není optimální řešení, jelikož by mohla databáze velmi nabývat na objemu. Druhé řešení bylo v použití typu blob, což je datový typ, který v sobě uchovává blíže neurčená binární data. Tento způsob je již o něco lepší. Objekt se slevami jsme tedy serializovali.

3.3 Dotazník

Popis

Tento projekt spočíval ve vytvoření desktopové aplikace, která při spuštění zobrazí uživateli formulář na vyplnění. Program měl také umět uložit vyplněná data, nahrát je při opětovném zapnutí programu znovu pro případnou editaci, vymazat uložené dotazníky nebo všechny uložená data hromadně uložit do předem dané struktury XML dokumentu. Zadavatel tady nekladal žádné specifické požadavky na použité technologie.

Řešení

Pro napsání programu jsem se rozhodl použít objektový jazyk C#, který je mi celkově ze všech naučených jazyků na vysoké škole nejbližší. Oceňuji hlavně jeho dobře napsanou dokumentaci, v které se mi oproti jazyku Java lépe hledá a orientuje. K práci jsem používal vývojové prostředí Microsoft Visual Studio 2010. Nejdříve jsem vložil všechny požadované prvky pomocí Designeru. Jsou to: menu, textové pole, rozbalovací seznamy, zaškrťovací seznamy a tlačítko pro potvrzení formuláře. Následovně jsem do grafických prvků nahrál všechna potřebná data.

Nyní popíši všechny možnosti v menu a jejich řešení. Nový formulář pouze smazal všechna vyplněná data v současném formuláři. K uložení dat jsem vytvořil metodu, která vytvořila složku saves, pokud ještě nebyla vytvořena. Nejdříve jsem chtěl data ukládat do Temp složky na disku s operačním systémem. Má to výhodu v tom, že spustitelný soubor může být umístěn kdekoliv na disku, načítání starých souborů to neovlivní. Po konzultaci se zaměstnavatelem jsem tento koncept změnil a dal jsem složku vytvářet na pozici, kde se spouští exe soubor aplikace. Toto na druhou stranu nese výhodu v tom, že neznalý uživatel přesně ví, kde se mu formuláře ukládají. Navíc nemůže dojít k tomu, že někdo při snaze vyčistit si svůj počítač, smaže všechna data v Temp složce spolu s uloženými pozicemi. Pro uložení formuláře jsem vytvořil třídu, která obsahuje všechna data v jednom objektu. Ten jsem pak serializoval do souboru, tedy převedl na posloupnost bitů.

Dále jsou v menu položky pro smazání nebo nahrání formuláře pro jeho následnou úpravu. Pro obojí jsem vytvořil nové rozbalovací okno, kde si může uživatel zvolit z uložených pozic, kterou chce smazat nebo uložit. Po smazání jsem musel vyřešit problém s načítáním uložených pozic. Ty byly totiž automaticky pojmenovávány od jedničky výše. Když například uživatel vymazal uložený formulář 2.save a existoval už 3.save, musel systém na něj při obětovném načtení přijít. To jsem poměrně jednoduchým algoritmem napravil.

A nakonec menu obsahuje položky pro uložení formulářů do XML souboru. Uživatel si v novém okně může ze zaškrťavacího seznamu vybrat všechny vyplněné formuláře, které chce exportovat. Následně si může vytvořit nový XML soubor nebo doplnit formuláře do již existujícího souboru na jeho konec. K tomu jsem využil třídu XmlWriter, která mi práci s XML soubory ulehčila.

3.4 Rezervační systém pro ubytovnu Pocheň

Popis

Mým posledním velkým projektem bylo vytvoření rezervačního systému pro ubytovnu ve vesnici s názvem Pocheň. Aplikace běží na stolním počítači, kde osoba na recepci má možnost přijímat objednávky, editovat je nebo smazat. Dále je možné pracovat se zákazníky (editovat, aj.) nebo pracovat s jednotlivými pokoji.

Při návrhu systému se muselo brát ohled na to, že ubytovna obsahuje více samostatných pokojů. Důraz je kladen na to, aby uživatel systému měl hlavní funkci, což je objednání nového pokoje, hned při spuštění před sebou. Se systémem se nepočítá, že by měl obsahovat velké množství přenášených dat jak do databáze, tak i zpět. Spouštění bude probíhat na obyčejném operačním systému Windows. Nebyl zde kladen požadavek na multiplatformnost.

Řešení

Stejně jako u projektu Dotazník jsem pro implementaci použil objektový jazyk C# a vývojové prostředí Microsoft Visual Studio 2010. Jako databázový systém jsem si zvolil MySQL.

Informační systém jsem se rozhodl napsat v třívrstvé architektuře především proto, aby šlo v případě potřeby napsat další prezentační vrstvu, třeba pro potřeby rezervace přes webové stránky. Pro každou vrstvu jsem vytvořil v projektu samostatnou složku. Mnou navrhovaná architektura obsahuje vrstvy:

- **prezentační** - zobrazení informačního systému uživatelům
- **aplikační** - obsahuje logiku aplikace
- **databázová** - zajišťuje propojení s databází

Nejdříve jsem navrhl databázi a pak její tabulky. Po několika náčrtech na papír jsem vytvořil tabulky Zákazníci, Objednávky a Pokoje a vzájemně je propojil vazbami. Následně jsem tabulky naplnil testovacími daty. Všechny SQL příkazy si vždy píši ručně a dělám si jejich zálohu do textového souboru. Pro kontrolu používám aplikaci phpMyAdmin.

Poznámka 3.1 Při programování normálně používám anglické názvy pro proměnné, metody, třídy, atd. Pro názvy v databázi jsem použil česká slova, což se mi v průběhu projektu neosvědčilo a několikrát mě to spleťlo.

Pro práci s databází jsem si musel do vývojového prostředí stáhnout a nainstalovat ovladač. Následně jsem ho přidal do projektu jako referenci. Nejdříve jsem vyzkoušel naprogramovat jednoduchou vazbu mezi vrstvami architektury. Vytvořil jsem tedy v Designeru menu a zde položku pro vytvoření nového zákazníka. Po kliknutí se zobrazí nové okno s formulářem, kde uživatel vyplní potřebné údaje a formulář odešle. Tato událost se dostane do aplikační vrstvy a následně do databázové, kde se údaj pošle do databáze jako SQL příkaz.

Po této úspěšné zkoušce jsem napsal téměř celou databázovou třídu, která obsahuje metody pro inicializaci k databázi, otevření a zavření spojení, vybrání, uložení, úpravu a smazání dat v databázi. Pro vybírání dat jsem použil generický typ List, který se skládal z dalších Listů. Vnořené listy již obsahují data z jednotlivých sloupců. Když například aplikační vrstva potřebovala znát tabulku se zákazníky, zavolala metodu Select s dvěma parametry. Ty obsahovaly string pro jméno tabulky a pole stringů s názvy sloupců. Pro lepší představu přikládám ukázkou zdrojového kódu.

```
public List<string>[] Select(string tableName, string[] columns)
{
    int numberOfColumns = columns.Length;
    string query = "SELECT_*_FROM_" + tableName;
    List<string>[] list = new List<string>[numberOfColumns];

    for (int i = 0; i < numberOfColumns ; i++)
    {
        list [i] = new List<string>();
    }

    if (this.OpenConnection() == true)
    {
        ...
    }
    return list ;
}
```

Výpis 1: Ukázka metody pro načítání dat z databáze

Při přidávání každé nové funkcionality programu jsem postupoval sestupně od vrchní vrstvy dolů. Nejprve jsem přidal do GUI požadované komponenty a nastavil jim potřebné vlastnosti. V některých speciálních případech jsem použil další formulářové okno. Následně jsem provedené akce nebo vyplněná data poslal do aplikační vrstvy, kde se zpracovávají. Podle potřeby jsem upravil třídu s databázovou vrstvou.

Základní obrazovka aplikace obsahuje menu, kde správce najde potřebné položky jako například editaci uživatelů. Dále správce vidí komponentu kalendáře monthCalendar. Zde se po kliknutí na vybraný den zobrazí tabulka. Každý jednotlivý sloupec reprezentuje jeden pokoj či apartmán. V následujících řádcích se zobrazí dodatečné informace jako například počet postelí nebo jestli pokoj obsahuje televizi. Na posledním řádku nalezneme stav, zda je pokoj volný, a možnost ho na daný den objednat.

Po kliknutí na objednávku je ještě nutné zadat potřebné informace do textových polí. Pole s názvem objednávaného pokoje se již doplní automaticky. Informace o zákazníkovi se mohou vyplnit dvěma způsoby. V případě, že se jedná o nového zákazníka, musíme formulář vyplnit ručně (navíc se přidá nový zákazník do tabulky zákazníků). Pokud je již klient v systému, lze ho vyhledat přes vysouvací menu. V tomto případě se vše vyplní automaticky. Nyní lze již odeslat objednávku a ta se projeví v databázi.

Aby se předešlo zbytečným přístupům do databáze, implementoval jsem v aplikační vrstvě aktualizaci dat tak, aby se již nemusely znovu načítat. To znamená, že například

při editaci zákazníka se v aplikační vrstvě edituje daná instance a do databáze se pošle jen příkaz pro editaci. Znovu načítání zákazníků již není potřeba.

Při psaní projektu, jako u každého jiného, jsem se často vracel k již napsaným částem a znovu je přepisoval. Například při zobrazení informací o pokojích a jejich následném objednání jsem původně použil komponentu `dataGridView`. Ta se ukázala jako nevhodná a musel jsem ji předělávat na `tableLayoutPanel`, kde se navíc jednotlivé komponenty daleko lépe vkládaly.

3.5 Nasazení webové prezentace

Popis

Tento projekt se týkal webové prezentace vzdělávací akce, která slouží nejen k předávání informací, ale i k přihlášení účastníků. Mým úkolem bylo převést dosavadní webové stránky vytvořené ve WordPressu [11] na nový informační systém. Zároveň jsem musel přesunout všechny potřebné texty a upravit nové prvky, aby odpovídaly grafickému návrhu.

Řešení

Nejdříve jsem se seznámil s informačním systémem. Ten je vytvořený za pomoci PHP a frameworku Nette [10]. Framework používá šablonovací systém Latte, který jsem si nastudoval. Poté jsem měl již všechny potřebné znalosti k převedení grafiky. Pro účely vývoje a testování mi byl systém nahrán na zkušební server. Zde jsem měnil čtyři CSS soubory a dva šablonovací Latte soubory tak, aby odpovídaly daným požadavkům.

V jednom úkonu jsem si musel přidat javascriptový soubor se skriptem. Editace některých elementů totiž ani pomocí šablonovacího systému nebyla možná.

4 Závěr

4.1 Uplatněné znalosti ze studia na vysoké škole

Jelikož jsem na praxi programoval, nejvíce jsem využil znalosti získané v předmětech *Základy programování I a II*. Při práci na informačním systému jsem využil na předmět *Vývoj informačních systémů*, který dobře spojuje více povinných i nepovinných předmětů probíhých během studia. Během prací na webových službách jsem využil zkušenosti, které jsme probírali v předmětu *Vývoj internetových aplikací*. Užitečný mně byl také předmět *Základy softwarového inženýrství*, když po mně firma vyžadovala návrhy průběhů určitého uživatelského postupu.

4.2 Získané znalosti při praxi

Během praxe jsem se musel naučit orientovat ve zdrojových kódech, které napsal někdo jiný. Žádné společné projekty jsme na vysoké škole nedělali, což je myslím škoda. Prohloubil jsem si znalosti ve vývoji informačních systémů a webových prezentací. Navíc jsem se musel naučit pracovat s některými programy, které firma používala. Třeba TortoiseSVN systém pro správu a verzování zdrojových kódů.

Jistě nejvíce zkušeností mi dala spolupráce na projektu Laňka a jeho widgetu Slev. Zjistil jsem jak programátor při tvoření myslí. Spolupracovník, který se mnou programoval v páru, neustále komentoval svoji práci a probíral se mnou řešení daného problému. Navíc jsem se mohl na každou nejasnost zeptat.

4.3 Scházející znalosti v průběhu praxe

Mezi znalosti, se kterými jsem se ještě nesetkal, patří již zmíněné verzování zdrojových kódů. Stávalo se mi, že jsem neznal méně používané specifikace daného jazyka. Jako příklad mohu třeba zmínit vnitřní třídy v Javě. Víím, že jsme je v *Programovacích jazycích I* brali, ale neznal jsem jejich smysluplné využití.

Problém jsem měl u znalosti pokročilejších programovacích technik. Mnohokrát jsem dlouhou dobu hledal určité způsoby řešení na internetu nebo posléze jsem se zeptal spolupracovníků na pracovišti. Celkově však cítím, že mi vysoká škola dala slušné znalosti, které jsem při této praxi využil.

4.4 Zhodnocení praxe

Při absolvování bakalářské praxe ve firmě YELLEN solutions s.r.o. jsem získal hodně nových zkušeností. Vývoj rozsáhlých systémů není tak jednoduchý a probrat vše během studia by asi nebylo možné. Spolupráce se zaměstnanci byla pro mne přínosná a motivující. Při problémech jsem měl možnost se zeptat a všichni se mi snažili vyjít vstříc a pomoci.

Byla to má první zkušenost s prací v kolektivu profesionálních programátorů. Myslím si, že by bylo přínosem pro všechny studenty zavedení povinné praxe už během studia na vysokých školách.

5 Reference

- [1] YELLEN solution s.r.o. [online]. [cit. 2013-04-28].
Dostupné z: <http://www.yellen.eu>
- [2] Redmine [online]. [cit. 2013-04-28].
Dostupné z: <http://www.redmine.org/>
- [3] Cufón [online]. [cit. 2013-04-28].
Dostupné z: <http://cufon.shoqolate.com/>
- [4] Web Developer [online]. [cit. 2013-04-28].
Dostupné z: <http://chrispederick.com/work/web-developer/>
- [5] Google Plugine for Eclipse [online]. [cit. 2013-04-28].
Dostupné z: <https://developers.google.com/eclipse/>
- [6] Google Plugine for Eclipse [online]. [cit. 2013-04-28].
Dostupné z: <http://php.net/>
- [7] jQuery [online]. [cit. 2013-04-28].
Dostupné z: <http://jquery.com/>
- [8] Java [online]. [cit. 2013-04-28].
Dostupné z: <http://www.java.com/en/>
- [9] C# [online]. [cit. 2013-04-28].
Dostupné z: <http://msdn.microsoft.com/en-us/vstudio/hh341490.aspx>
- [10] Nette [online]. [cit. 2013-04-28].
Dostupné z: <http://nette.org/cs/>
- [11] Wordpress [online]. [cit. 2013-04-28].
Dostupné z: <http://wordpress.org/>